Quantum Linearization Attacks

Xavier Bonnetain[1], Gaëtan Leurent[2], María Naya-Plasencia[2], André Schrottenloher[3]

[1] Université de Lorraine, CNRS, Inria, Nancy, France
[2] Inria, Paris, France
[3] Cryptology group, CWI, Amsterdam, The Netherlands

Outline

Introduction

Quantum attacks in symmetric crypto

Q1 model

- Make classical queries to the black-box (e.g. cipher)
- Do quantum computations

$\Longrightarrow$ application of Grover's algorithm to key-recovery.

$\Longrightarrow$ realistic, less powerful.

Only **polynomial** speedups (so far) in symmetric crypto.

Q2 model

- Do quantum computations
- Can query the black-box **inside the quantum algorithm**, as a **quantum / superposition oracle**

$\Longrightarrow$ theoretical, strictly more powerful, but non trivial.

$\Longrightarrow$ perhaps more suited for provable security.

Exponential speedups become possible.

Quantum superposition attacks

Quantum hidden shift algorithms break **some symmetric constructions** that are classically secure, using **superposition queries**.

1. These attacks use the **structure** of the construction, independently of its subcomponents;
2. They **translate** a key / state-recovery problem into a **hidden shift / period problem**;
3. They solve the problem using an off-the-shelf **quantum hidden shift / period algorithm**;
4. The algorithm requires **superposition queries** to the construction.

Typical period-finding tool: Simon's algorithm

Simon's problem

Let $f : \{0,1\}^n \to \{0,1\}^n$ be a function, either injective, or two-to-one with a period: $f(x) = f(y) \iff y = x \oplus s$; determine the case.

Simon's algorithm

- There is a procedure that, with a **single superposition query to f**, samples a random y such that $y \cdot s = 0$
- ("Superposition query": $\sum_x |x\rangle |0\rangle \mapsto \sum_x |x\rangle |f(x)\rangle$)
- With $\mathcal{O}(n)$ queries and $\mathcal{O}(n^3)$ time, we can find s / determine if f is periodic
- (Works also if f is a random function)

Simon, "On the power of quantum computation", FOCS 1994

Our result

- The **quantum linearization attack**: a new way to use period-finding to target symmetric constructions
- We use it to create polynomial-time (superposition) **forgery attacks** on many MAC constructions, using Simon's algorithm (but also Deutsch's, Bernstein-Vazirani, and Shor's)

Targets

- Parallel MACs (and BBB variants): ΘCB3, LightMAC, LightMAC+, Deoxys, ZMAC, PMAC_TBC3k, PolyMAC, GCM-SIV2
- XOR-MACs, MACs based on universal hashing
- New superposition attack on Poly1305 with about 32 queries

Quantum Forgery Attacks on MACs

MACs

A function:

$$\begin{cases} \mathrm{MAC}_k : \{0,1\}^\nu \times \{0,1\}^* \to \{0,1\}^t \\ \qquad\quad IV \quad , \quad m \mapsto \mathrm{MAC}_k(m) \end{cases}$$

which produces an **authentication tag** under a given secret key k.

Existential unforgeability (EUF-CMA)

Under **chosen message queries**, the adversary shouldn't be able to output a valid $\{\mathrm{IV}, \mathrm{message}, \mathrm{tag}\}$ pair for a new {IV, message}.

IVs are chosen at random, or not repeated (nonces).
We can consider MACs with or without them.

Quantum security notions

Since the adversary makes superposition MAC queries, what is a "new" message in the EUF-CMA definition?

- **PO-unforgeability [BZ13]:** An adversary making q MAC queries shouldn't be able to output $q+1$ valid $\{\text{message}, \text{tag}\}$.
- **BU [AMRS20]:** An adversary making "blinded" MAC queries shouldn't be able to output the tag of a message that is blinded.
- **qPRF security:** An adversary making queries shouldn't be able to distinguish MAC_k, for a random k, from a random function.

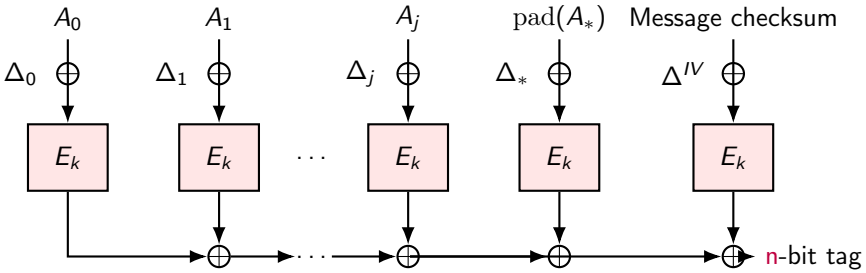
It is known that $qPRF \implies BU \implies PO$, and **we will break PO.**

Boneh and Zhandry, "Quantum-Secure Message Authentication Codes", EUROCRYPT 2013

Alagic, Majenz, Russell, Song, "Quantum-Access-Secure Message Authentication via Blind-Unforgeability", EUROCRYPT 2020

Example: OCB

OCB (here version 3) is a block cipher-based AEAD. We consider the (IV-based) tag computation, with associated data blocks $A_0, \ldots, A_j, A_*$.



- The offsets $\Delta_0, \Delta_1, \ldots, \Delta_j, \Delta^{IV}$ should remain secret
- The IV remains classical and **changes at each query**

Krovetz, Rogaway, "The Software Performance of Authenticated-Encryption Modes", FSE 2011

Example: OCB (ctd.)

With an empty message and two n-bit blocks $A_0 = x, A_1 = x$, we have:

$$\mathrm{MAC}_k(IV, A_0, A_1) = F_{k,IV} \oplus E_k(\Delta_0 \oplus A_0) \oplus E_k(\Delta_1 \oplus A_1) \ ,$$

which means that:

$$\mathrm{MAC}_k(IV, x, x) = \mathrm{MAC}_k(IV, x \oplus \mathrm{s}, x \oplus \mathrm{s}) \ ,$$

where $\mathrm{s} = \Delta_0 \oplus \Delta_1$.

Although the IV changes at each query, Simon's subroutine **uses a single query:** what matters is that s stays the same.

Knowing s:

- $A_0 \| A_1 \| \ldots$ and $A_1 \oplus \mathrm{s} \| A_0 \oplus \mathrm{s} \| \ldots$ always have the same tag.
- we can produce two valid $\{\text{message}, \text{IV}, \text{tag}\}$ from a single query

Summary of previous attacks

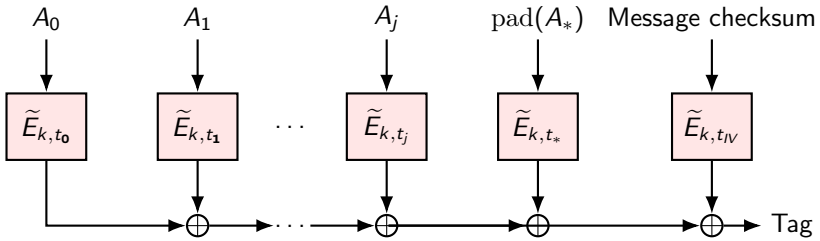
When a controlled value (e.g., message block) is XORed to a secret value (e.g., internal state), we can:

- embed a **hidden boolean period / shift** in a MAC query
- recover it with Simon's algorithm
- then double the number of tags produced by MAC queries

The Quantum Linearization Attack

Example: ΘCB

ΘCB is like OCB, but based on a **tweakable** block cipher: the use of tweaks renders the block cipher calls independent.



- The previous attack based on Simon's algorithm does not work
- But there is another simpler attack **based on Deutsch's algorithm**

Example: ΘCB (ctd.)

Deutsch's algorithm

Decide if $f : \{0,1\} \mapsto \{0,1\}$ is **constant** or **balanced** with a **single superposition query**, i.e., compute $f(0) \oplus f(1)$ with a single query.

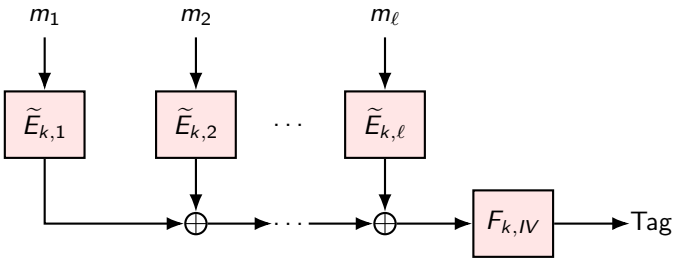
We take a single block $x \in \{0,1\}$: $\mathsf{MAC}_k(IV, x) = F_{k,IV} \oplus \widetilde{E}_{k,t_0}(x)$, and truncate the output to a single bit.

- With a single query to MAC_k, we obtain the bits of $\widetilde{E}_{k,t_0}(0) \oplus \widetilde{E}_{k,t_0}(1)$ and reconstruct the value.
- Querying the tag of any $0\|A_1\| \ldots$, we can forge the tag of $1\|A_1\| \ldots$

Deutsch's algorithm bypasses the XOR with $F_{k,IV}$, which was a sufficient protection classically.

New example

- Let's replace the final $\oplus$ by post-processing by a function F
- Such a construction, with or without IVs, yields classically secure MACs such as LightMAC and PMAC



New result: Simon's algorithm strikes again.

The new idea

We **restrict the inputs** so that each block takes only two values: $m_1 = b_1||0, \ldots, m_\ell = b_\ell||0$ and make a function of an ℓ-bit input:

$$\mathrm{MAC}_{k,IV}(m_1, \ldots, m_\ell) = G_{k,IV}(x) = G_{k,IV}(b_1|| \cdots ||b_\ell) = F_{k,IV}\left(\bigoplus_{1\le i\le \ell} \widetilde{E}_{k,i}(b_i||0)\right)$$

$H : x \mapsto \bigoplus_i \widetilde{E}_{k,i}(b_i||0)$ is an affine function of x. Indeed:

$$H(x) = \bigoplus_{1\le i\le \ell}\left(b_i \odot \left(\widetilde{E}_{k,i}(0) \oplus \widetilde{E}_{k,i}(1)\right) \oplus \widetilde{E}_{k,i}(0)\right)$$

$$= \underbrace{\begin{pmatrix}(\widetilde{E}_{k,1}(0) \oplus \widetilde{E}_{k,1}(1)) & \cdots & (\widetilde{E}_{k,\ell}(0) \oplus \widetilde{E}_{k,\ell}(1))\end{pmatrix}}_{M_\ell\text{: binary matrix, n rows and } \ell \text{ columns}} \times \begin{pmatrix} b_1 \\ \ldots \\ b_\ell \end{pmatrix} \oplus \bigoplus_i \widetilde{E}_{k,i}(0) \ .$$

The new idea (ctd.)

When $\ell \geq n + 1$, the kernel of M_ℓ is non-trivial. Each of its elements α is an ℓ-bit string such that:

$$\forall x, H(x \oplus \alpha) = H(x) \ ,$$

and so:

$$G_{k,IV}(x) = F_{k,IV}(H(x)) = G_{k,IV}(x \oplus \alpha) \ .$$

We can recover such an α with Simon's algorithm.

- Data complexity: $\mathcal{O}\left(n^2\right)$, with $\mathcal{O}(n)$ queries of $\mathcal{O}(n)$ blocks
- Time complexity: $\mathcal{O}\left(n^3\right)$ to compute α
- Once it is obtained, we can proceed as before (produce two valid tags for a single query)

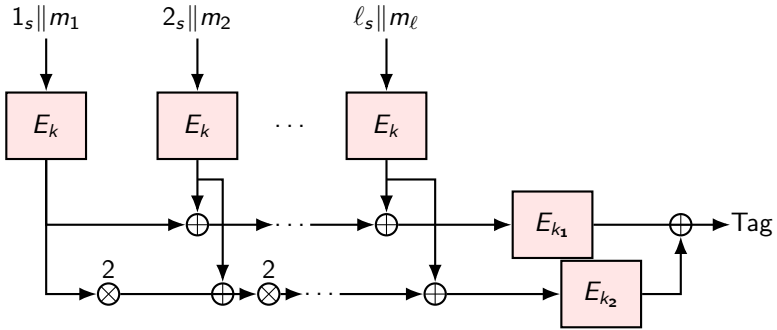
Attacks on some BBB MACs

- There have been many proposals for "beyond birthday-bound" (BBB) secure MACs, which offer more than $n/2$ bits of security against forgery attacks.
- Double Hash-then-Sum: message blocks are independently processed in two pipes of size n, and the results are XORed.

Because there are two independent processes, the "standard" Simon-based attacks reach only a complexity $\widetilde{\mathcal{O}}\left(2^{n/2}\right)$. [GWHY21]

Guo, Wang, Hu, Ye, "Attack Beyond-Birthday-Bound MACs in Quantum Setting" PQCrypto 2021

Example: LightMAC+



Naito, "Blockcipher-Based MACs: Beyond the Birthday Bound Without Message Length", ASIACRYPT 2017

LightMAC+, abstracted

The input $m_1, \ldots, m_\ell$ is processed with independent TBC calls, then combined in two different ways:

$$\mathrm{MAC}_k : (m_1, \ldots, m_\ell) \mapsto f_k\left(\bigoplus_i \widetilde{E}_{k,i}(m_i), \bigoplus_i 2^i \widetilde{E}_{k,i}(m_i)\right) .$$

There are **two** matrices M_ℓ, M'_ℓ and two column vectors C, C' such that:

$$\mathrm{MAC}_k(b_1||0, \ldots, b_\ell||0) := f_k\left(M_\ell \times \begin{pmatrix} b_1 \\ \vdots \\ b_\ell \end{pmatrix} \oplus C, M'_\ell \times \begin{pmatrix} b_1 \\ \vdots \\ b_\ell \end{pmatrix} \oplus C'\right) ,$$

Then as soon as $\ell \geq 2\mathsf{n} + 1$, there is a non-trivial vector α such that:
$M_\ell \alpha = M'_\ell \alpha = 0$, and so, there are periods.

Where does it stop?

We have polynomial-time attacks on LightMAC, LightMAC+, PolyMAC, GCM-SIV2, Deoxys, PMAC+, ZMAC... and any construction that:

- processes the input blocks **independently**
- computes one or more XOR-linear functions of these processed input blocks
- computes the tag from the outputs of these functions

Thus PO-unforgeability requires either:

- some sequentiality (e.g., NMAC) (**this works**)
- IV-based constructions that process all the blocks with the IV (**this works**)
- using a $+$ instead of $\oplus$ (**we don't know if this works**)

Conclusion

Conclusion

Quantum cryptanalysis perspective

- The "quantum linearization" attack is a new, generic attack based on Simon's algorithm, fundamentally different from the previous ones.
- It breaks many MACs, and practically all deterministic parallelizable constructions proposed so far.

Provable security perspective

Can we design efficient parallelizable qPRFs (implying quantum-secure MACs)?

Concrete security perspective

Does this have consequences when the queries are only classical?

Thank you!